

my Raspberry Pi

HELP

IF YOU ARE ADMIN AND WANT SETUP THIS WEB SITE YOU NEED TO KNOW:

this little project is a mix of PHYTON FLASK example FLASKR and a HTML template

pls see links in footer.

added work for theme and blog function /edit/delete/..

have fun to start from there!

there is no install requirements for RPI (as with RASPBIAN 2/2017)

python3 , flask 0.10.x, sqlite3 (no CLI) ready

pls see -q- if you want use Arduino Firmata USB connection

-a- for blogging must login first

user: **admin**

password: **default**

this password is now encrypted in the database see

/menu/ SITE SETUP

-b- in menu BLOGEDIT

can ADD new blog entries

or change existing (modify / delete / quit) by the small edit icon at each existing entry

also from detail window can call now the EDIT if logged in

-c- later can erase existing html pages

this help.html

and elements.html from **menu.html**

-d- pls change impressum

/menu/ SITE SETUP

-e- in the menu have default shortcuts Mx

pls edit in menu.html to call special blog entries (by index / you see in editblog list) and give some catching text

-f- you sure not like my colors?

pls see /myblog/static/assets/css/main.css and use editor find "KLL" so you see the main changes i made for the theme (color).

-g- in header.html line 11 and 25

i use 2 images for the favicon.ico and the header logo banner.png, pls change,

pictures are (to be) at /static/images/
instead of changing header.html and copy now pictures, can ju
overwrite the picture files using old names "favicon.ico" and
"banner.png"

-h- in index.html need some headline see
/menu/ SITE SETUP

-i- the database is /static/data/myblog.db
there is also the schema.sql file.
for reset or make new database file (with different tables ..) see
at end of **flask_app.py** line 561: # init_db() and un-comment it (for
one run)
now this is also the only way to reset user password.

-j- add a gallery page / menu entry
now also from a database of pictures and remarks

-k- add setup page
to edit lots of SITE settings (now in database)
be careful when change user , password / there is no way back /
as password is now encrypted

-l- add button [Backup] (DB as sql) to SITE SETTINGS
/myblog/static/data/Backup_schema.sql

-m- add HTML editor "ONLINE tiny MCE" for "textarea"

in files editblog.html (ADD) and modblog.html (EDIT)

-n- picture file uploader

a basic uploader of pictures to the /static/images/ directory

and in the Blog can be called (by Tiny MCE image icon)

source: **../static/images/mypicture.jpg**

-o- gallery in database

and a gallery picture selector page

incl remark text area edit

-p- a page views and operates RPI GPIO

/menu/ RPI GPIO

channed definition in **rpi_gpio.py**

use BCM

out_chan_list = [23,24,25]

in_chan_list = [17,27,22]

(HTML auto update 10sec)

-q- a page views and operates Arduino I/O (Firmata version)

/menu/ RPI ARDUINO Firmata

expecting that ARDUINO UNO is loaded with (new) Standard Firmata

and connected by USB

if not the page will fail

pls install on RPI for python: **sudo pip3 install pyfirmata**

channed definition in **rpi_arduino.py**

```
Dout_chan_list = [4,13]
```

```
Din_chan_list = [8,12]
```

```
Aout_chan_list = [3,5] # D out PWM UNO: 3,5,6,9,10,11
```

```
Ain_chan_list = [0,1] # A in UNO: 0 .. 5
```

(HTML auto update 10sec)

-r- now use Arduino UNO board with own PID .. control code
and a custom JSON readable (but slow) protocol.

for test making face plate (groups) of uPCSweb loops (TAGs)

/menu/ FP test

if Arduino is loaded with project code uPCS2 (see included sketch)

/menu/ FP group

and from there with click on faceplate

FP_detail/x (where x == DBI)

OPEN: process overview picture also to call FP_group FP_detail

at the very first start a ram disk file is created "/run/shm/uno_dbi.dat"

see also get_DBI_list in file rpi_upcs2.py

with the static uPCS2 lopp point table

at that moment also a "init_run_UNO.sh" is called what will run (or
first install)

a python service what uses the USB interface to Arduino UNO and
makes the required communication.

the live data are feed to the same ramdisk file for the FP_... html
pages to show.

-s- clean up some things:

there was a footer and footer short (/templates/"file".html)

now there is only a footer what imports footer_social.html

but only if the showSocial is set true from flask_app.py:

```
"return render_template('index.html', entries=entries, setting=setting,
showSocial=True)"
```

temporary FP_* use custom header and footer for the jquery and other JS calls

now for modblog.html and editblog.html use ", showBBC = True" a switch to enable tinyMCE in "header.html"

and for data / hardware related pages use ", autoupdate = True" as switch in "header.html" to enable 10sec auto reload

in rpi_arduino.html, rpi_gpio.html, FP_UI.html, FP_group.html, FP_detail.html

faceplate_svg.html now calculates the line pix positions from

PV,SP,OV, alarms inside, using like

$y1 = \{ \{ 40 + 2 * (100 - dbi.ALL) \} \}$, but alarm lines are disabled when 0.0% or 100.0%

but ranged values like

$$\text{DBI_list}[\text{aindex}][\text{'RPV'}] = \text{DBI_list}[\text{aindex}][\text{'LOR'}] + (\text{DBI_list}[\text{aindex}][\text{'HIR'}] - \text{DBI_list}[\text{aindex}][\text{'LOR'}]) * \text{DBI_list}[\text{aindex}][\text{'PV'}] / 100.0$$

$$\text{DBI_list}[\text{aindex}][\text{'RSP'}] = \text{DBI_list}[\text{aindex}][\text{'LOR'}] + (\text{DBI_list}[\text{aindex}][\text{'HIR'}] - \text{DBI_list}[\text{aindex}][\text{'LOR'}]) * \text{DBI_list}[\text{aindex}][\text{'SP'}] / 100.0$$
 still calculated in "def calc_SVG_DBI_list(DBI_list):" and handed down to template to SVG by dbi.RPV as added columns.

-t- the uPCS background service program

upcs2_service.py uses same rpi_upcs2.py code and definition of a loop point database.

so actually it does not matter what starts first, the ram disk file is made.

it also makes a trending to 2 floatingpoint matrix

PID1 timestamp, PV SP OV MD (stored at PID1 PV update)

INDx timestamp, IND1PV, IND2PV, IND3PV, IND4PV (stored at IND1 PV update)

at same point (every 5 sec) both arrays are exported to ramdisk to be used in a new FP_trend display

means (ringbuffer pointer logic) to make temporary JSON record list and that dumped to ram disk file.

-u- Historic Trend module

every 5 min a CSV type record of PID1 PV OV,

IND1,IND2,IND3,IND4,DCD1 PV Averages is stored to

/home/pi/media/USBstick/HDdb.csv (if not exist to /run/shm

/HDdb.csv)

a LINE CHART type trend (SVG 1000px)
and a CSV export HTML page (converted to real units) is create

-v- Alarming module
on PV alarms are checked according limits, and alarm texts and sys
log entries generated

-w- DCD1 loop point TYP M21
Motor 2 inputs, one output
Arduino D3 D4 D5 D6 are inputs, NOTFAIL / RUN / FIELD START / HW
INT
where first 2 generate a PV
Arduino D10 D11 D12 D13 are outputs, START / (OPEN) / (OPEN) /
(OPEN)
a software interlock from PID1_PVAHH locks SP MODE OUTPUT
(EXAMPLE)
the HW interlock into D6 can lock the mode to show (EXAMPLE
DISABLED)

- start? -
well, you read this so i expect that this site is up.
but there is now a file: "install.sh", start with
./install.sh it makes a file "start" you can call with
./start what does a
sudo python3 flask_app.py using YOUR path info.

after this you see that webserver starts at PORT 4567

so from any (remote) browser call **RPI_IP:4567**, to change at

flask_app.py line 13

for remote access pls open that port / port forward to RPI_IP / in your router.

the install also creates a icon at your Desktop to start the webserver.

- open? -

at rev 0.8.1 the mail is used with mailto: and some social links are connected to networks using raspberry pi accounts but a python mail tool (requires installation) still open.

- more info -

kll engineering

© CC BY SA | design: HTML5 UP | driven by: FI ASK
| adapted: KLL engineering